Documentation on efforts measurement

Year 2022-2023
Company : Sony CSL

Supervisor : Jérôme SZEWCZYK
Client : David Colliaux - Peter Hanappe
Engineers : Razane Azrou - Céline Zakhary - Wilson Crochet - Mathilde Vangeluwe

1. Structure

The client wants to measure the efforts generated by the robot. This structure is a two-directional sensor, it measures the horizontal efforts and vertical ones. It is composed (from bottom to top) of a squared-closed base, made of profiles, to delete any torque parasites. Then fixation, to fix three cells for vertical efforts, on which rest the cells - two of 5 kg capacity and one of 50 kg. Small profiles rest on them to put on a linear system of translation, to enable horizontal efforts. The system supports a wooden board fixed to the horizontal cell by an L-shaped fixation.

a. Modelisation

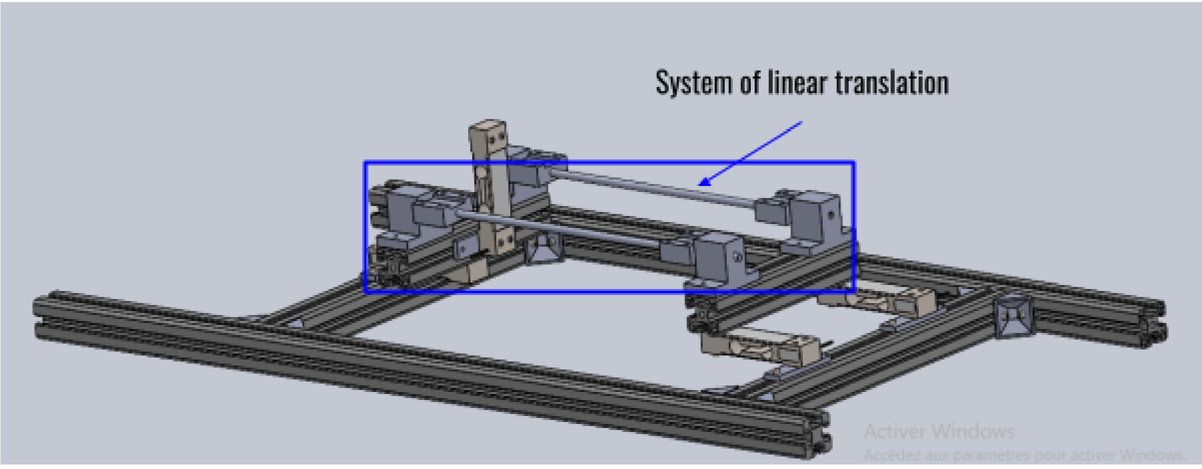


Figure 1 : Structure of efforts measurement

b. Conception

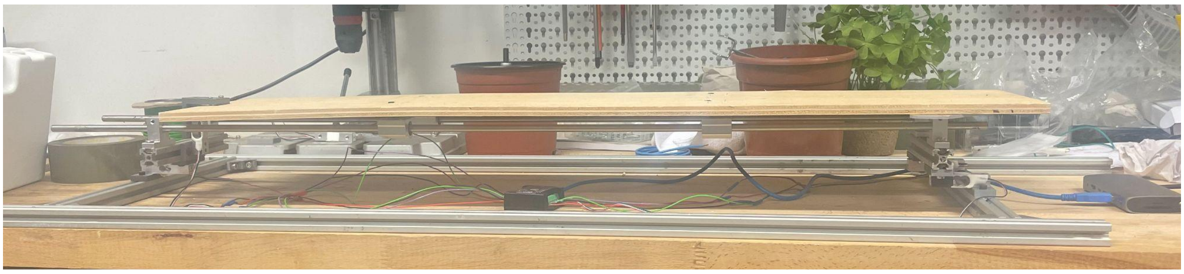


Figure 2 : Structure for efforts measures - side view

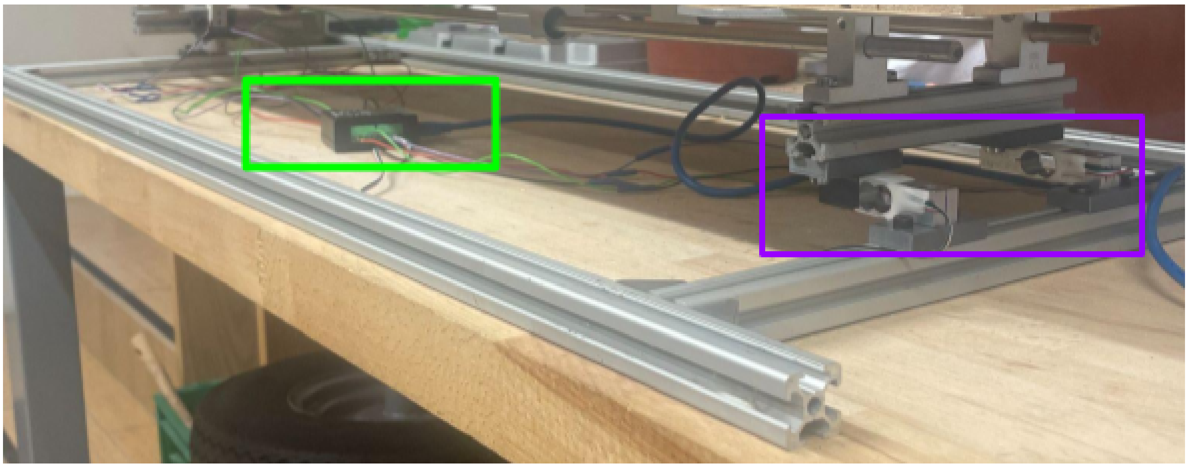


Figure 3 : Square based structure with profiles
Purple box : Vertical efforts measure cells - 5 kg each
Green box : Phidget 4-input bridge

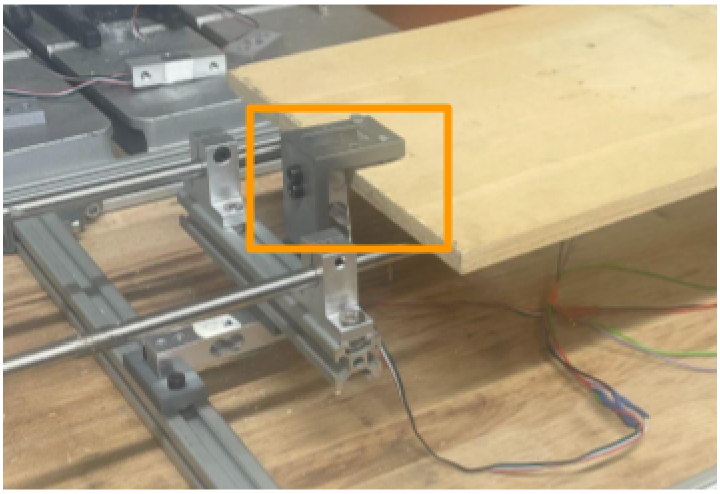


Figure 4 : left-side of the structure (view figure1)
Third vertical effort cell - 20 kg
Orange box : Horizontal effort cell - 780g + L-shaped fixation

Figure 6 : Structure - left front view

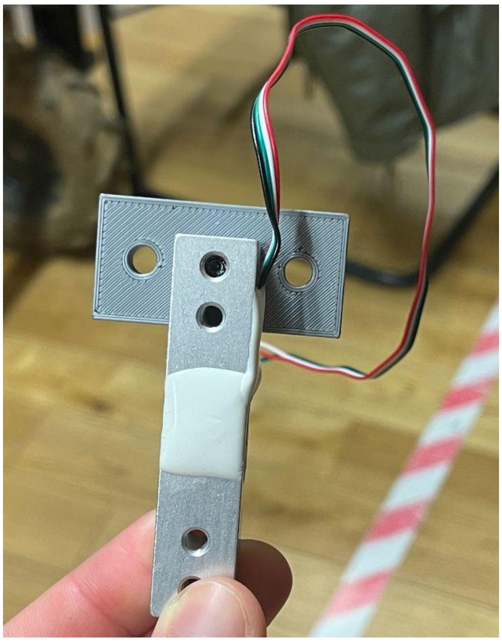


Figure 5 : Fixation of the cells on the profiles

2. Electronic

a. Material

- 4-input bridge Phidget (Wheatstone bridges)
- 4 load cells :

Name	Cell 5_1	Cell 5_2	Cell 20	Cell 780
Capacity (kg)	5	5	20	0.780
Nature of effort measured	Vertical	Vertical	Vertical	Horizontal

b. Connectivity

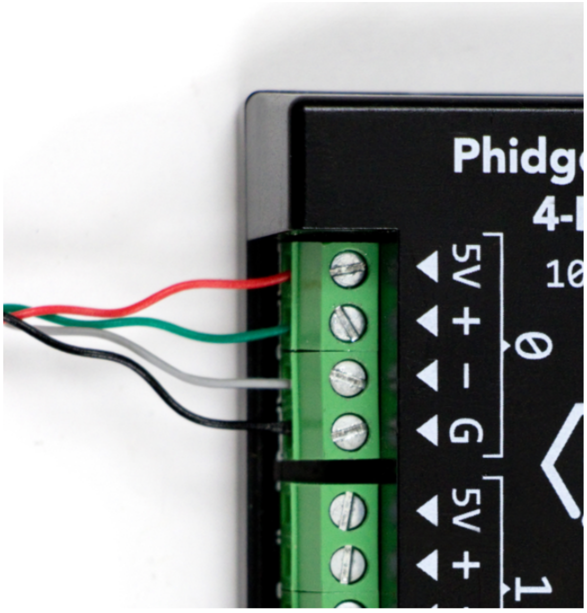


Figure 6 : How to connect the cells on the Phidget

- The cables were extended with welds to be able to attach the 4 cells. So the white cable is purple for 3 cells.
- Connecting the Phidget object to a laptop : a microusb - usb cable
- Attaching the cells as follow (It is mandatory, or the code won't work) :
    - cell 5_1 : input 1
    - cell 5_2 : input 0
    - cell 20 : input 3
    - cell 780 : input 2

3. **Computing**

a. Calibration of each cellule individually

i. CalibrationAnalyse.py : collect data

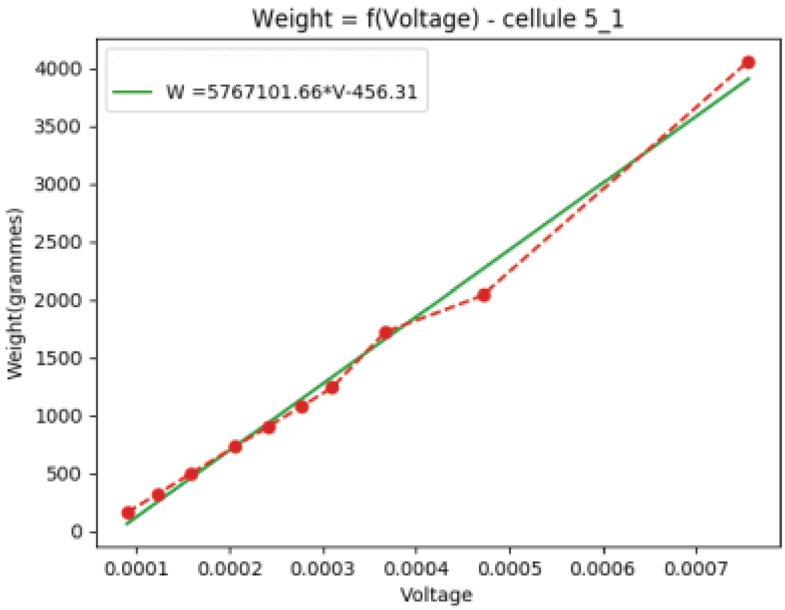
- Collect 1000 measures on 10 different objects
- Calculate standard deviation, the mean value of voltage measured
- Sorts
- Saves : weights entered, the mean value of voltage for each object and the standard deviation for each object (This data is saved in Text_files → VoltageMean – Weight – Samples files)

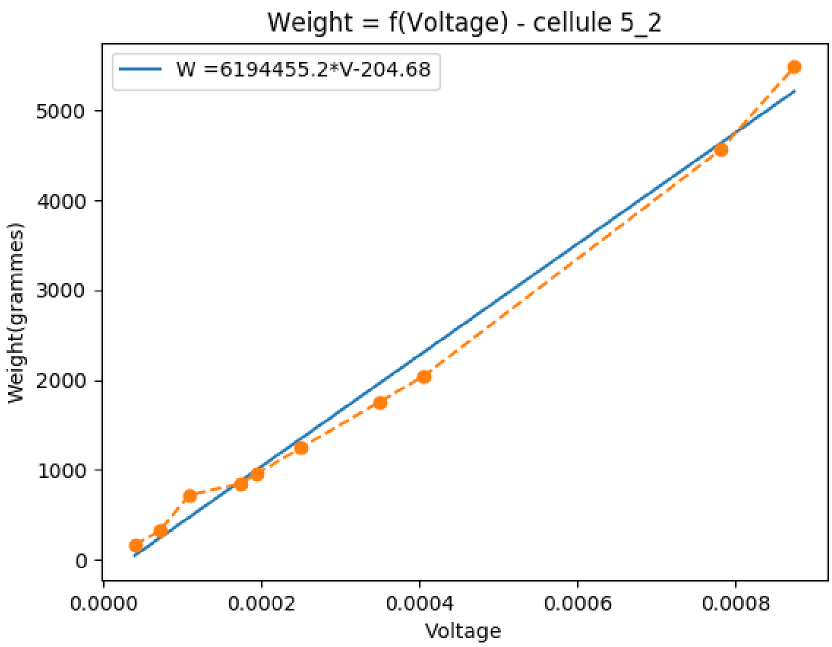
ii. *CalibrationProcess.py : analyze data*

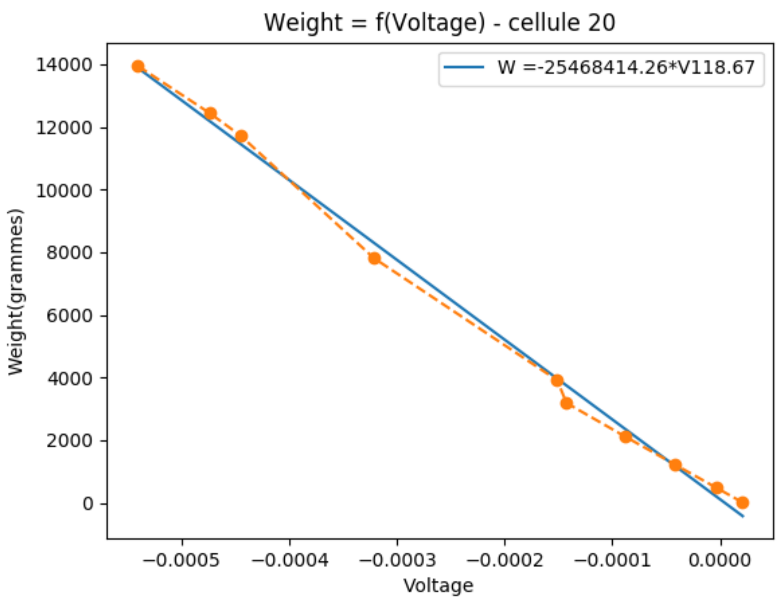
- Opens files and reads data to analyze
- Applies a linear regression : relation between weight and voltage

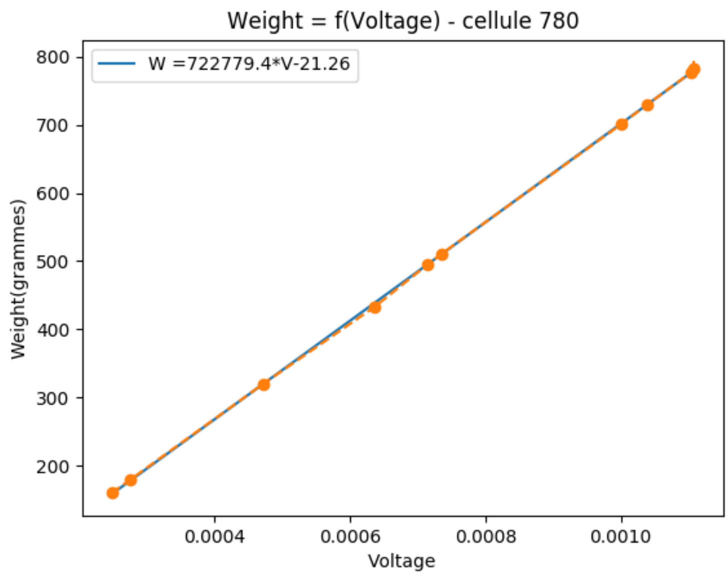
	Cell 5_1	Cell 5_2	Cell 20	Cell 780
Slope	5767101.655903426	6194455.19811738	-25468414.26472861	722779.3974552653
intercept	-456.3077783036904	-204.67628016617346	118.6727476709375	-21.2563001758013

- We plot the models




Weight = f(Voltage) - cellule 5_2
W =6194455.2*V-204.68
Weight(grammes)
0
1000
2000
3000
4000
5000
0.0000
0.0002
0.0004
0.0006
0.0008
Voltage


Weight = f(Voltage) - cellule 20
W =-25468414.26*V118.67
Weight(grammes)
0
2000
4000
6000
8000
10000
12000
14000
−0.0005
−0.0004
−0.0003
−0.0002
−0.0001
0.0000
Voltage



- We plot the standard deviation

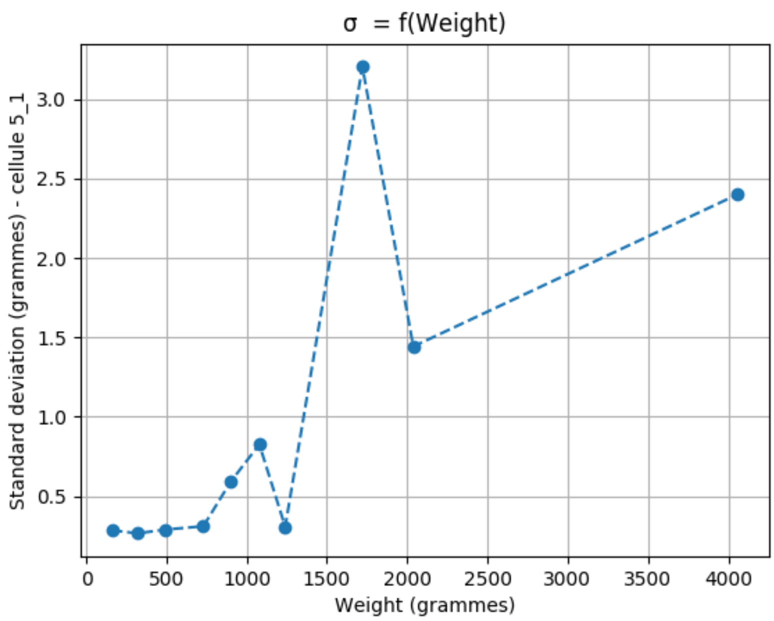


Figure 7 : Standard deviation - cell 5_1

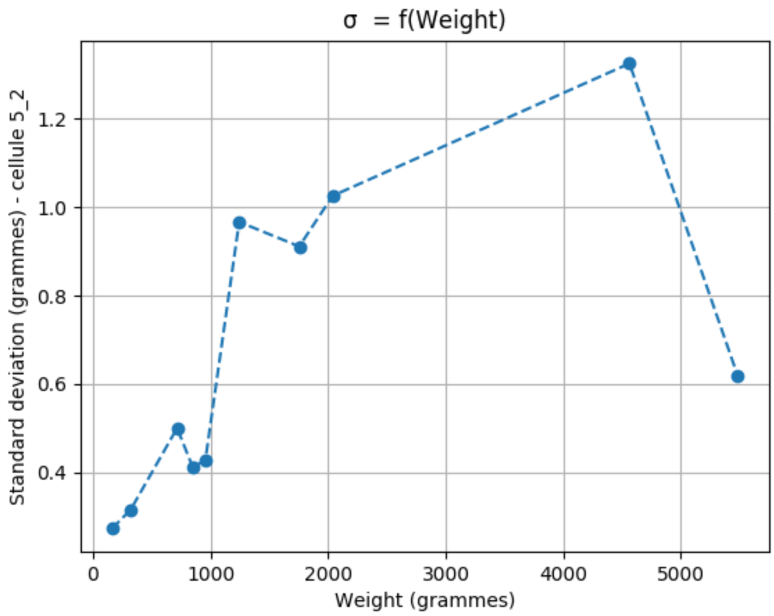


Figure 8 : Standard deviation - cell 5_2

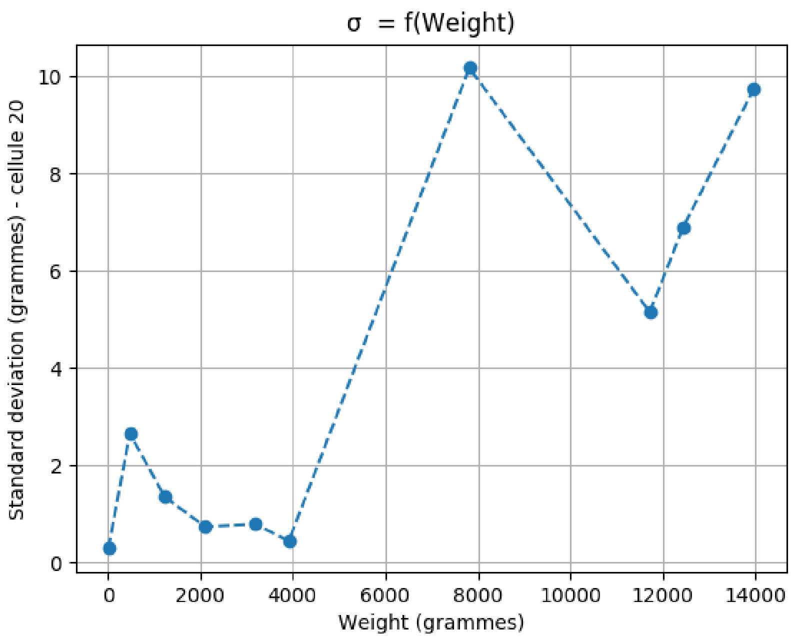


Figure 8 : Standard deviation - cell 20

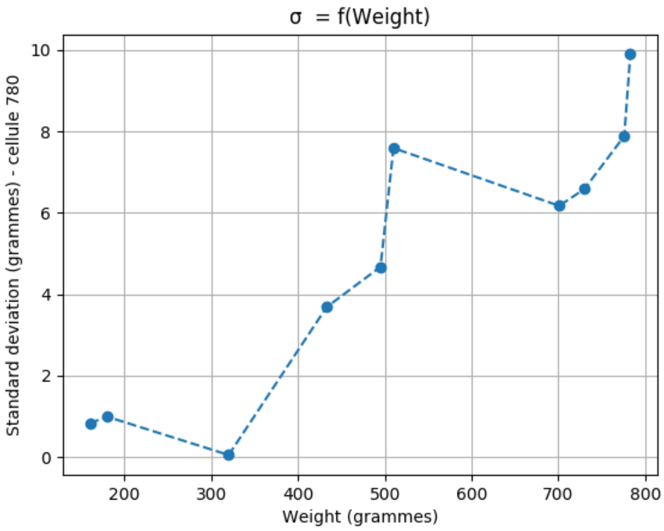


Figure 9 : Standard deviation - cell 780

b. Calibration of the structure for each direction

i. Horizontal direction

To calibrate the structure in this direction, I used a dynamometer : pushing on the cell with a known force. I compare the real efforts to the one measured :

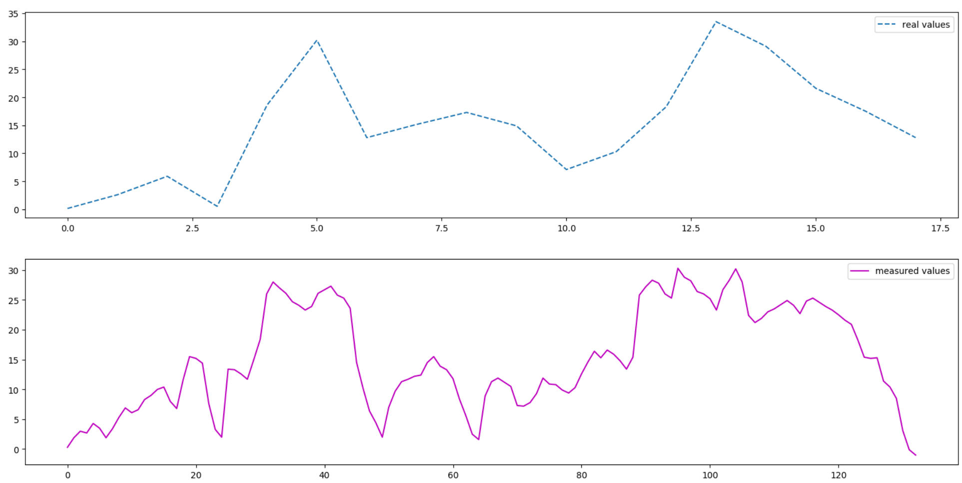


Figure 10 : Horizontal calibration using a dynamometer

We can see that the results are almost similar. Having more measured points than real ones, I had to plot them separately. But, we can use the cell 780 with confidence.

ii. Vertical direction

I used the same protocole than with the individual calibration. Only the number of samples is smaller. The result is satisfying, with an error of 3-4 N :

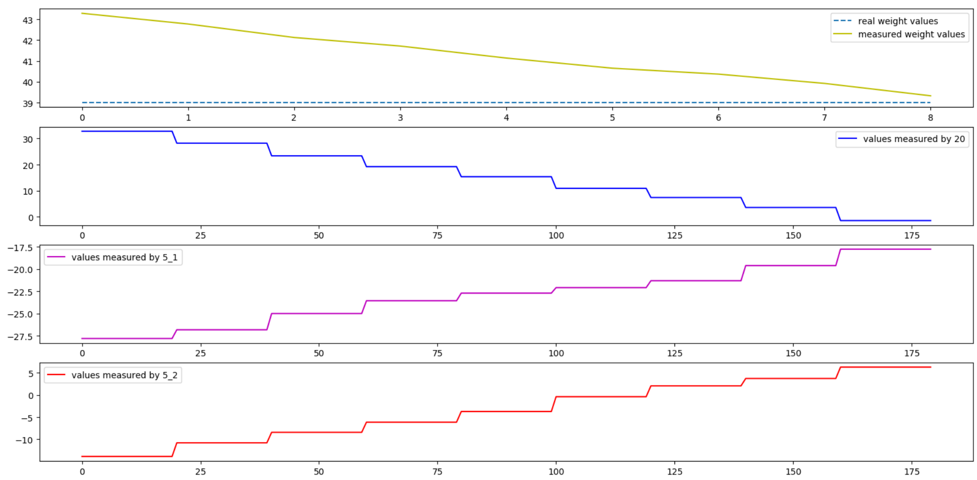


Figure 11 : Vertical calibration and each cell contribution

iii. A problem encountered

In the beginning I used a 25kg cell instead of the cell 20. However, while calibrating the structure I noticed that if I slide a weight from left to right the weight measured increases :

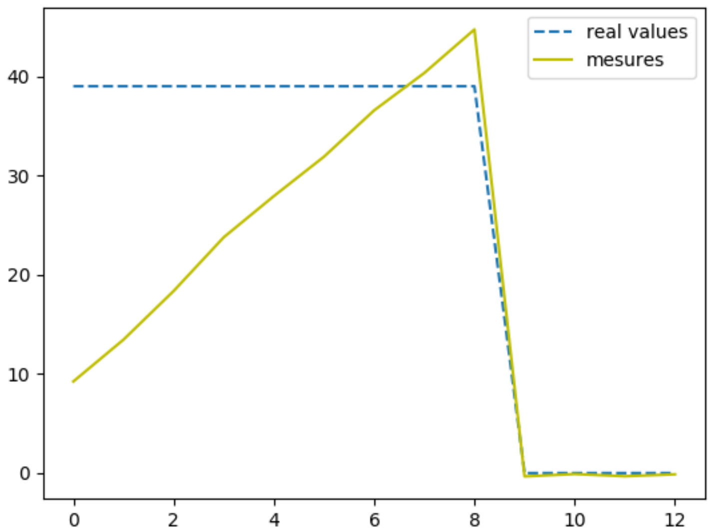


Figure 12 : Flop in the measures

Which was problematic, the structure being in this case unfunctional.
By displaying individually the results of each cell, I ascertained that the 25 kg capacity cell was dysfonctionnel.

c. Force_mesures_live.py : displays on live vertical and horizontal efforts in time

Goal :
Display in real time the efforts involved while weeding with the robot.
Details on code :

Global variables :

- Maximum period of time to display before refreshing = 10 s
- Maximum weight to measure = 35 kg
- Arrays of data : time and efforts (displayed and saved)
- Time
- Interval of time of frames displaying = 250 ms
- Phidget objects : 4 load cells

Functions

- *init()* : initializes the 2D object (*lines*)
- *ConvertVoltage(voltage0, voltage1, voltage2, voltage 3) :* converts a measured voltage to effort in N, using the results of linear regression. It also saves it in the right array of data.
    - *Voltage 0 : voltage measured by cell – 5_2*
    - *Voltage 1 : voltage measured by cell – 5_1*
    - *Voltage 2 : voltage measured by cell – 20*
    - *Voltage 3 : voltage measured by cell – 780*
    - NB : This order is very important or the result will be false
- *on_press(event)* : saves data after closing the window (event handler)
    - event : the event to handle
- *animate()* : Collect the measures, fills the 2D object, refreshes the figure to display on live the horizontal and vertical efforts. It also gives the time of acquisition between two measures.

Offsets of measures :

Horizontal charge = 7.82 N +- 3 N
Vertical charge = 21.3 N +- 3 N (approximately 0.1 N of precision)
NB : data is stored in *WeightExperiment_horizonta.txt and WeightExperiment_vertical.txt files.*

Time of acquisition between two measures

Time acquisition = 0.1 s

4. Experiences

a. Collecting data

i. Dry soil without using close looping

There were four experiences with a pot and a certain profile of the soil :

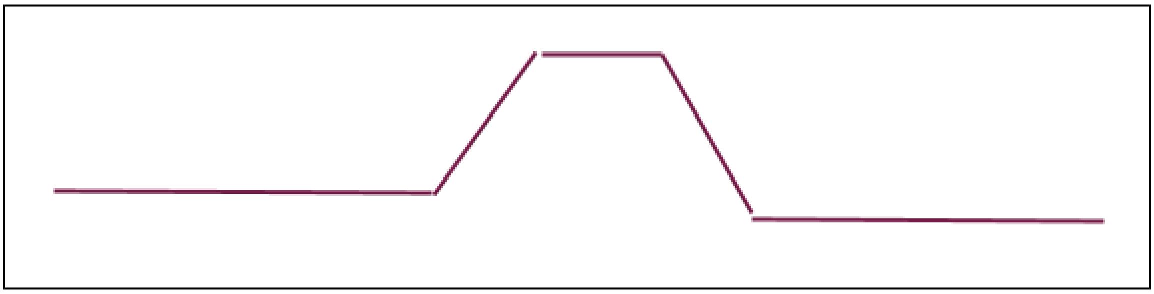


Figure 13 : first profil of the soil

And there is a video of the experience :

[First experience]

I collected the measured efforts individually of the current of the motors. Here are the profiles of each try :

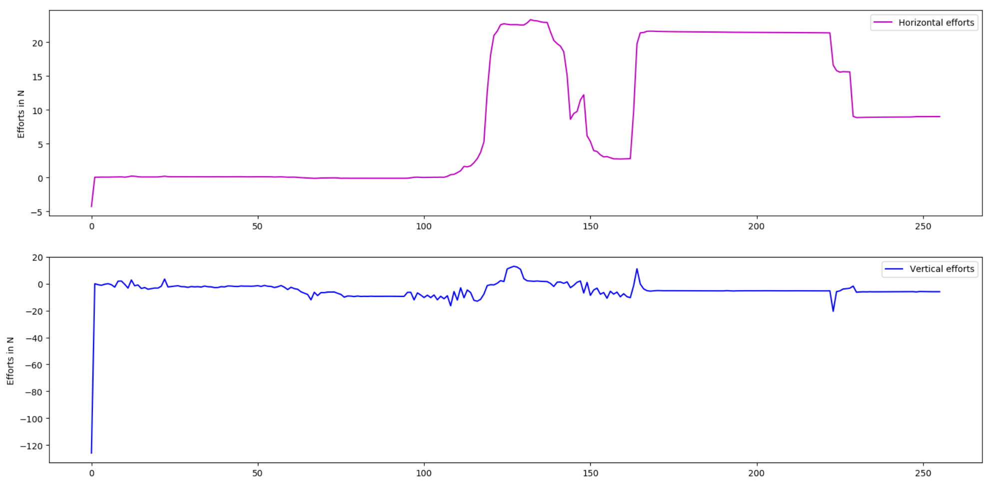


Figure 14 : First try

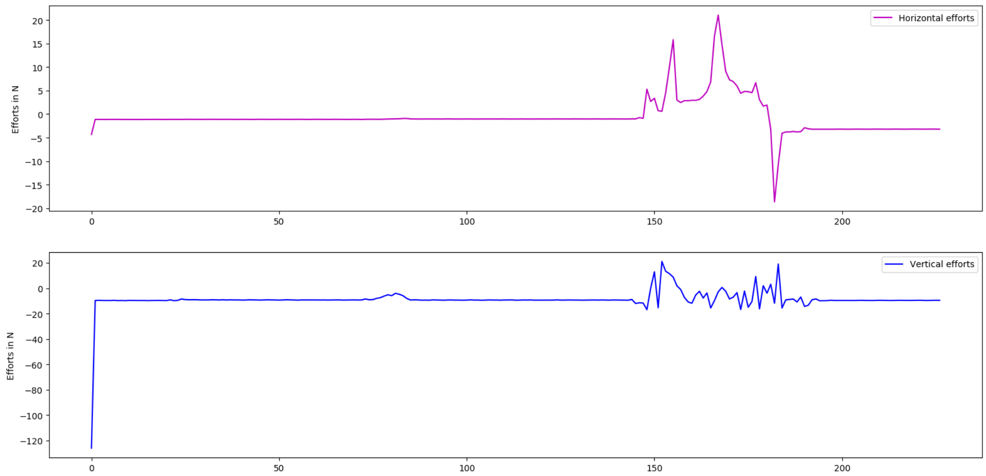


Figure 15 : Second try

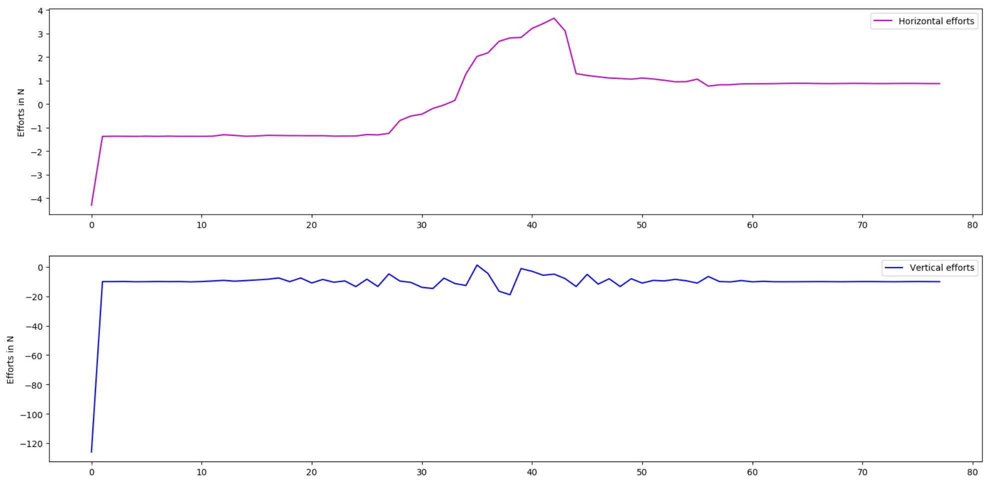


Figure 16 : Third try

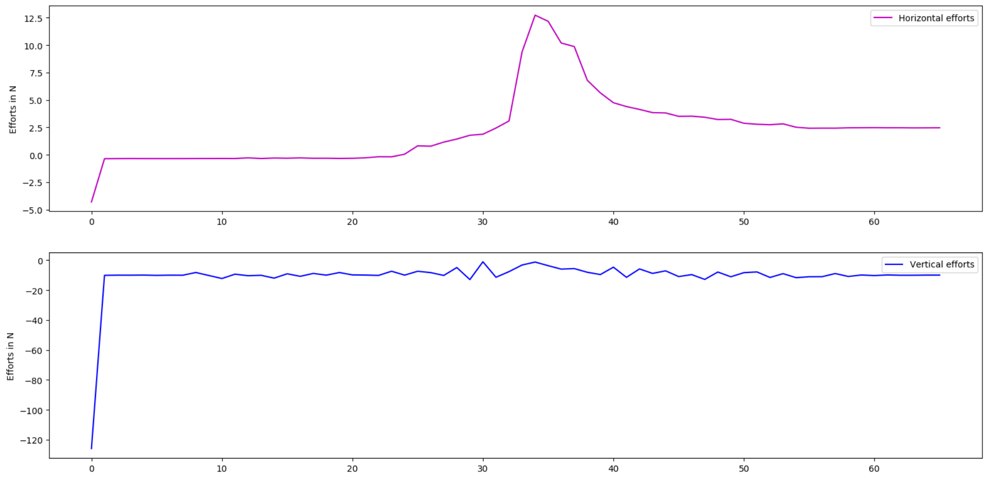


Figure 17 : Fourth try

We notice that the vertical effort isn't that important compared to horizontal efforts. But at that point I didn’t know how to use those efforts to control the weeding hand. A solution could be to plot the efforts as a function of the current, to deduce a model. I had to synchronize the starting of my acquisition with those of the current of the motors.

b. Synchronisation with current acquisition

For doing so we collected the CPU time start of each of us, then we subtracted the two to get the difference. I always start before Wilson, because sometimes the Phidget doesn’t respond. For the first experience the difference was of 2.301202297 seconds.
Wilson then sent me his measurements. However, while he got 6000 measures, I got only 135 ones. Which constitutes another problem. I then tried to change the code to find where the issue was.

c. Current-effort model

By displaying the time of each acquisition, we, me and Wilson, understood that the problem resided in the frequency of acquisition : the Phidget gots a frequency 40 times lower than of Odrive motors.
My time of acquisition was initially at *6.794929504394531e-6 seconds*, because I only calculated the mean value of the 10 first acquisitions. And because the 4 first acquisitions are fast before stabilizing around 0.1 seconds, the mean value was distorted.

Then to plot efforts as a function of current I calculate the mean value on 40 values, and restrain the array to be of the same length as the one of efforts.
It is done in the *Tracer_experience.py* script.
Current_fct_efforts_h_x.py script plots the desired outcome.
NB : the plots are as follow :

- Horizontal efforts as a function of the current of x axis and the current of the rotative weeding tool
- Vertical efforts as a function of the current of z axis and the current of the rotative weeding tool

We tried this on a second experience. Where the profile is as follow :

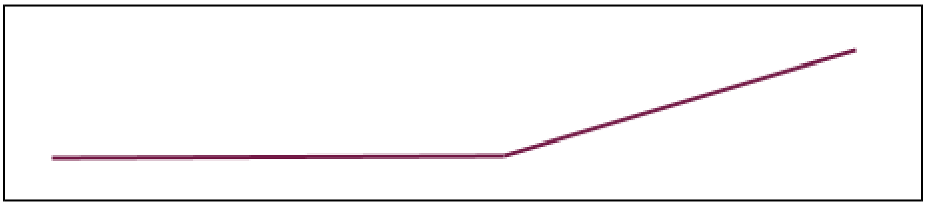


Figure 14 : Second profile of soil

But the results weren't exploitable. The noise was to importante and the profil of the current couldn't represent the slope, considered too light.

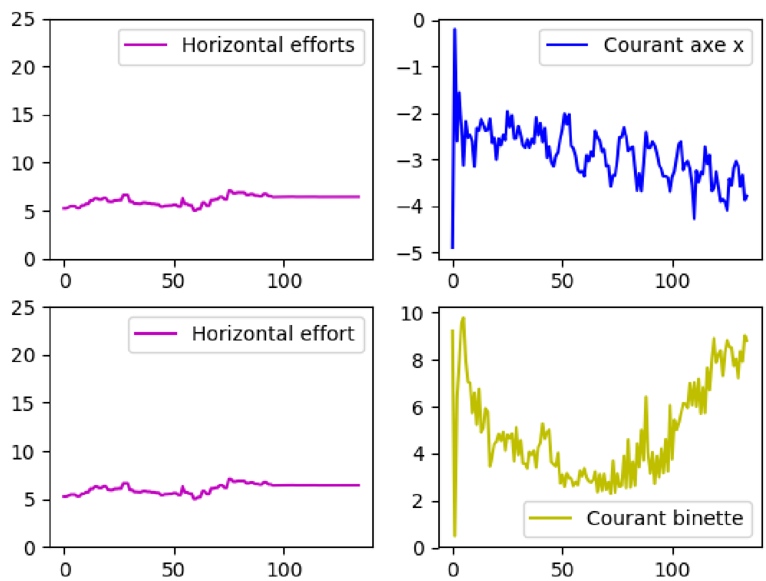


Figure 15 : Horizontal efforts - x axis and rotative tool currents

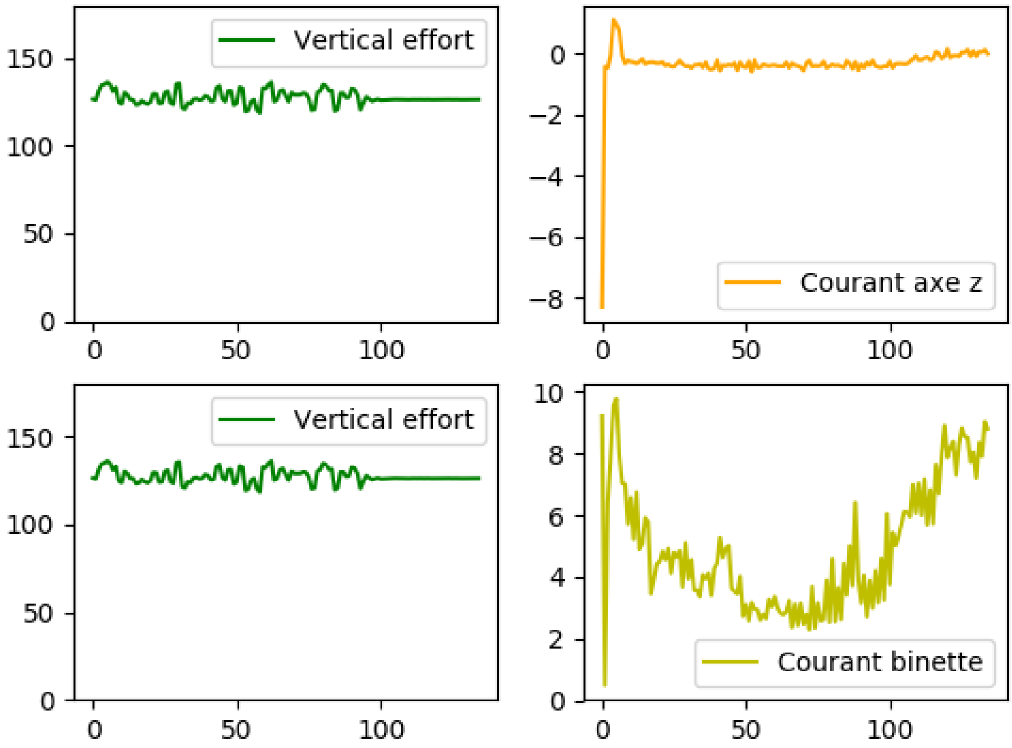


Figure 16: Vertical efforts - z axis and rotative tool currents

So it when plotting the desired function it looks more like children drawings :

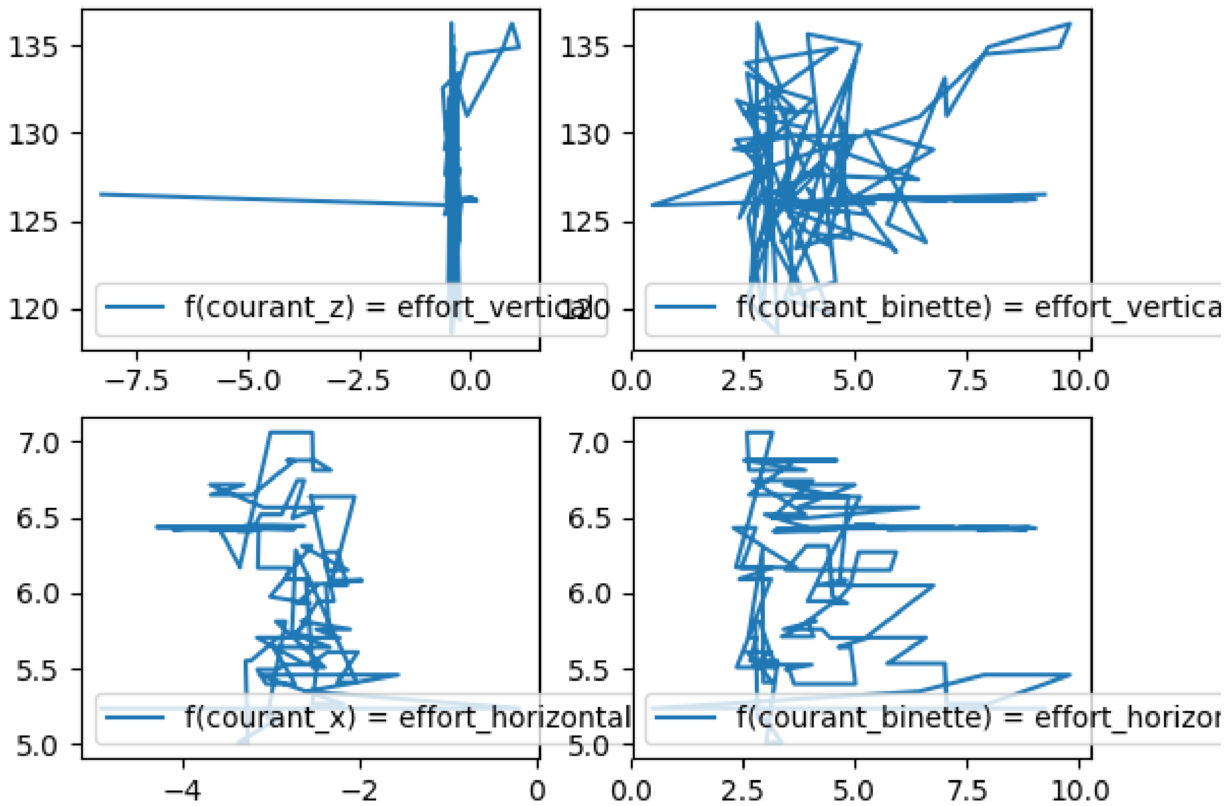
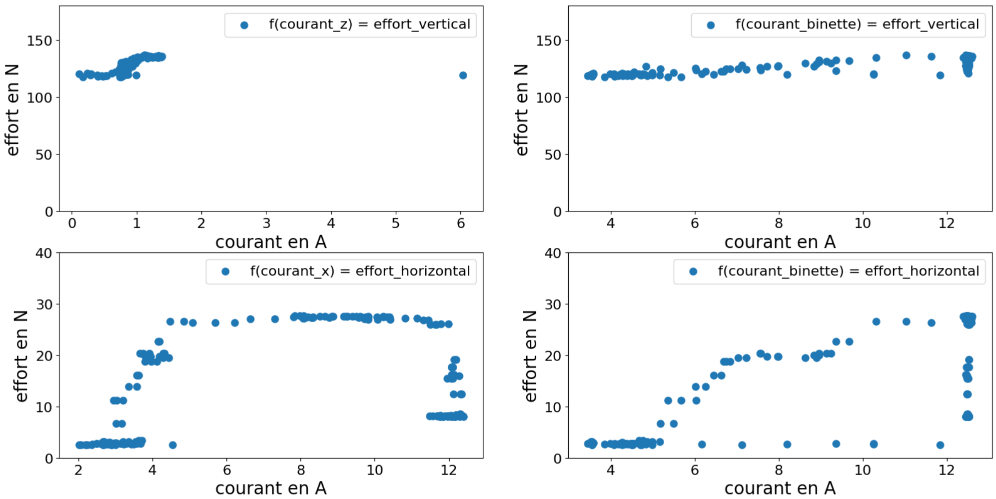
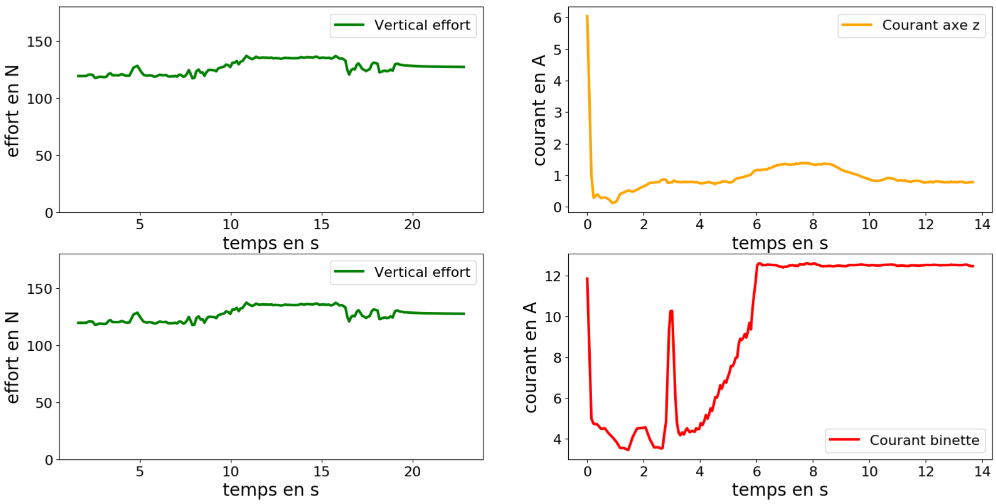


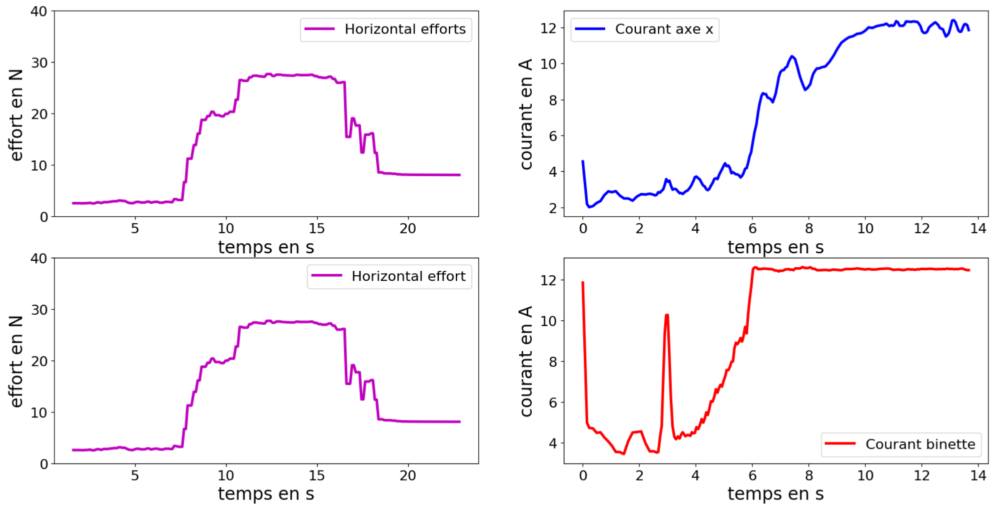
Figure 17 : Children drawings

We'll do a third experience on the 16th of may and see.

We then decide to repeat the first experiment 10 times, plot each of the efforts to see how they evolve through time. The currents and then the relations between the two.

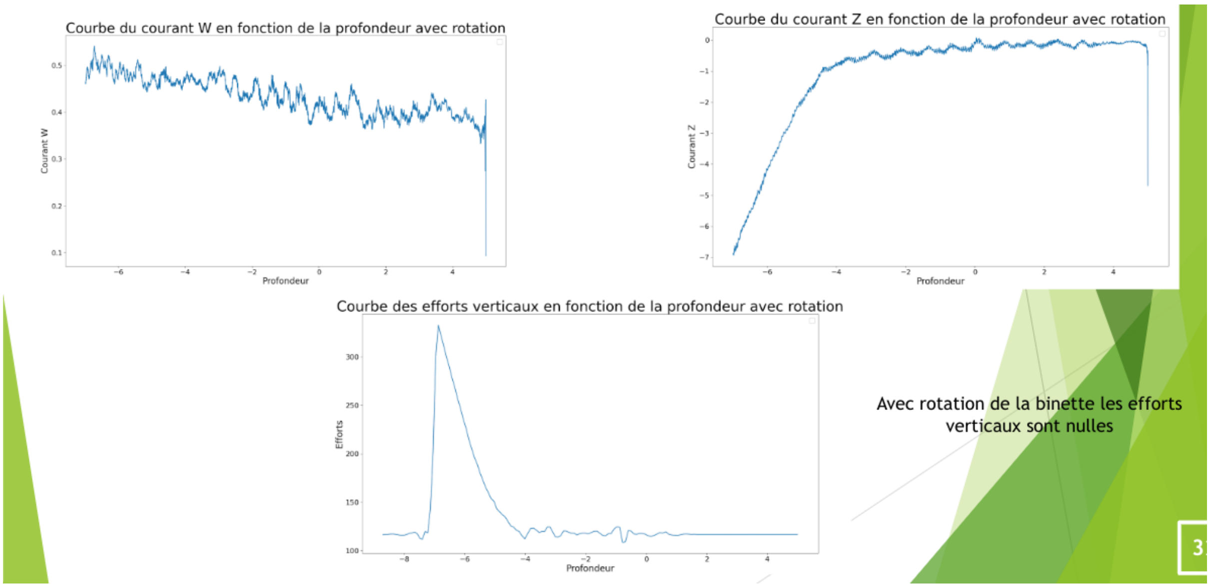

f(courant_z) = effort_vertical
f(courant_binette) = effort_vertical
f(courant_x) = effort_horizontal
f(courant_binette) = effort_horizontal
effort en N
courant en A


Vertical effort
Courant axe z
Vertical effort
Courant binette
effort en N
courant en A
temps en s



We can see that the vertical efforts couldn’t be taken into account, only the horizontal efforts vary. The best information to use for the control is the current of the weeding tool. The horizontal efforts are indeed showing a linear tendency with this current.

A last experience we made, to confirm that the vertical efforts aren’t exploitable, is the vary the position of the z axis and rotation the weeding tool :



We can see that while the weeding tool continues to rotate the vertical effort will stay almost equal to zero. It is raised once the rotation is blocked. So the vertical effort aren’t useful till the weeding tool is rotating, which will always be the case.

Conclusion :

In conclusion, the efforts show that our first guess of using the efforts or current of the x axis for control isn't right. The weeding tool's current being the best entry for the desired result.